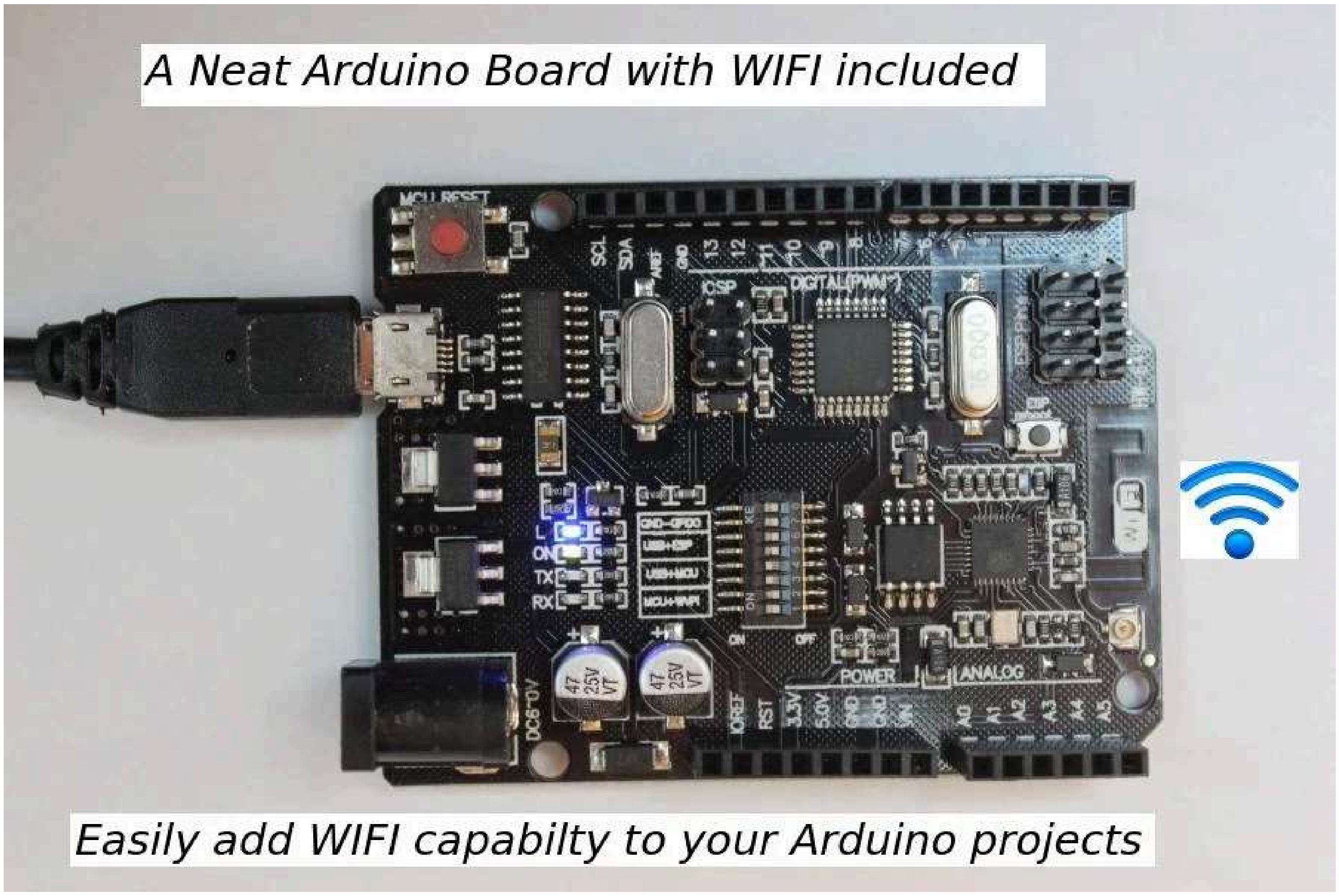
AUTODESK Instructables

UNO R3 + WIFI ESP8266 + CH340G Arduino and WIFI - a Versati le Development Board

By kevinjpower in CircuitsArduino


A Neat Arduino Board with WIFI included
Easily add WIFI capabilty to your Arduino projects

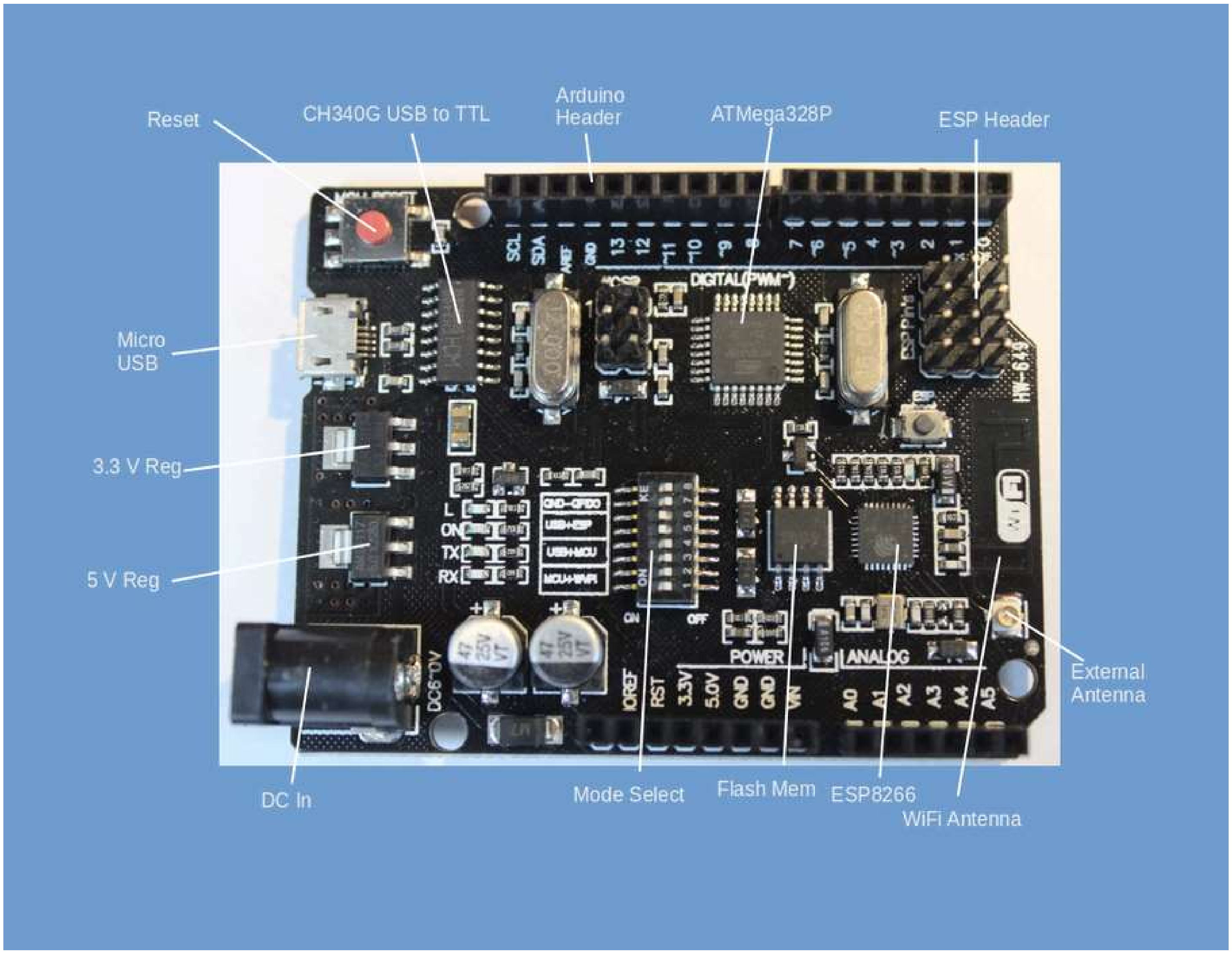
This project shows how to use the ATmega328 and ESP8266 single board manufactured by Robodyn that allows for wifi connectivity to an Arduino on a single development board.

This specific development board has numerous advantages (which will be covered later) but lacks co mprehensive documentation. This is surprising because it would seem the ideal development platform for WiFi enabled Arduino projects. Hopefully this project can help with providing more details.

Supplies

Arduino ESP8266 WIFI development board

Step 1: The Arduino ESP8266 WIFI Development Board



A search for the following text “UNO R3 WIFI ESP8266 ATMEGA328 CH340G” will yield numerous results of sites that sell this particular development board. It can be found on eBay, AliExpress, Banggood and other sites. Cost is approximately 10 to 14 USD before shipping. Before purchasing, make sure that the photograph in this article matches the picture on the selling site.

High level specifications of the board as follows:

- Model Number: UNO+WiFi-R3-AT328-ESP8266-32MB-CH340G
- Supply Voltage: 6 to 9V
- CPU: ATmega328
- WiFi: ESP8266
- USB- TTL converter: CH340G
- Header PinOut: Uno R3

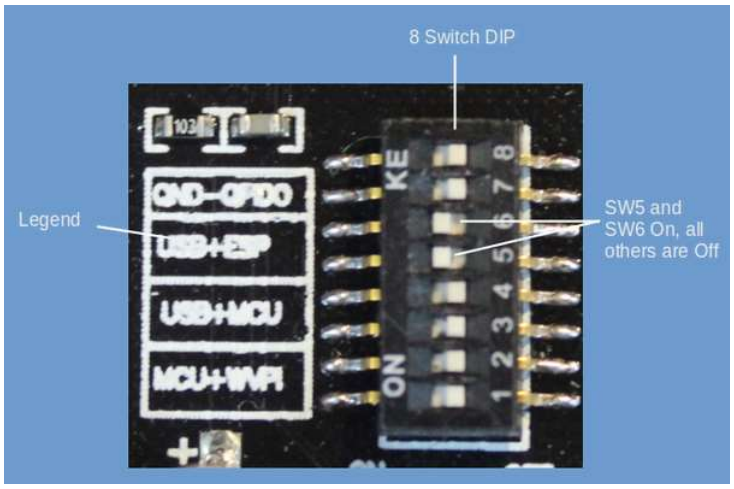
A description of the components on the board

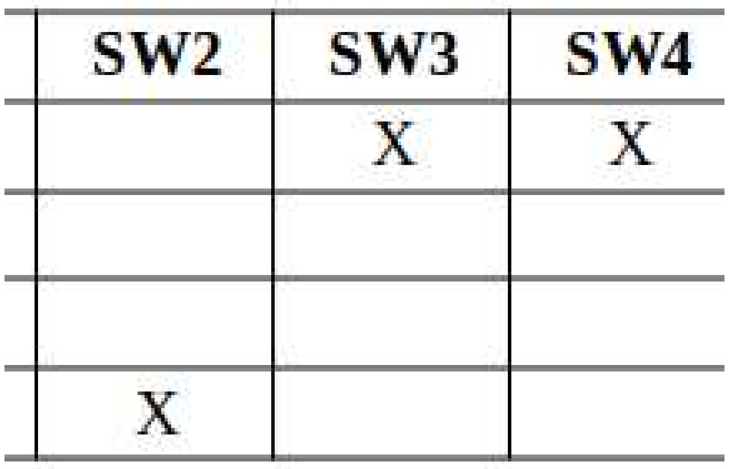
- The board includes a 5V and 3.3 V Voltage regulator to allow for the differing voltage requiremen ts for the ESP8266 and ATMega328
- Instead of the large USB connector used on a standard Arduino, this board uses a micro USB co nnector for serial communication and programming
- The Reset button resets both the ESP 8266 and ATMega328
- The header pin out is standard Arduino; this allows any Arduino compatible shield to be connecte d to this development board
- The board includes a small WiFi antenna. If greater range is required, provision is made for conn ecting an external antenna

It seems that this specific board has several advantages over similar solutions (specifically the WeMos D1 board)

- The Arduino pin out allows for connection of many of the Arduino shields available
- The different voltage levels are built into the board and become transparent to the User
- It allows for easy implementation of WiFi communication in Arduino projects
- As there are two processors incorporated on the board, a greater range of processing power is a vailable
- Communication between the ATMega328 and the ESP8266 is via the serial lines and utilizes the well known Serial library

Step 2: Mode Selection





SW2	SW3	SW4
	X	X
X		

Crucial to the operation of the board is the Mode Selection switch, consisting of a 8 switch DIP mounte d in the center of the board.

The mode switch allows for four possible settings

- Mode 1 – USB Serial connected to ATMega328 for both programming and serial communication
- Mode 2 – USB Serial connected to ESP8266 for programming
- Mode 3 – USB Serial connected to ESP8266 for serial communication
- Mode 4 – ATMega328 and ESP8266 serial connected together, allowing for serial communicatio n between the two devices. The voltage difference between the Arduino and ESP8266 is taken c are of in the board.

The ESP8266 requires that GPIO0 is pulled to ground while programming, a requirement not applicabl e to the ATMega328. Hence there are two separate modes for the ESP8266; switching SW7 on groun ds GPIO0.

The DIP switch settings for these modes in the table

DIP Switch settings for various modes. X = Switch On Blank = Switch Off

The board has a legend showing the switch settings

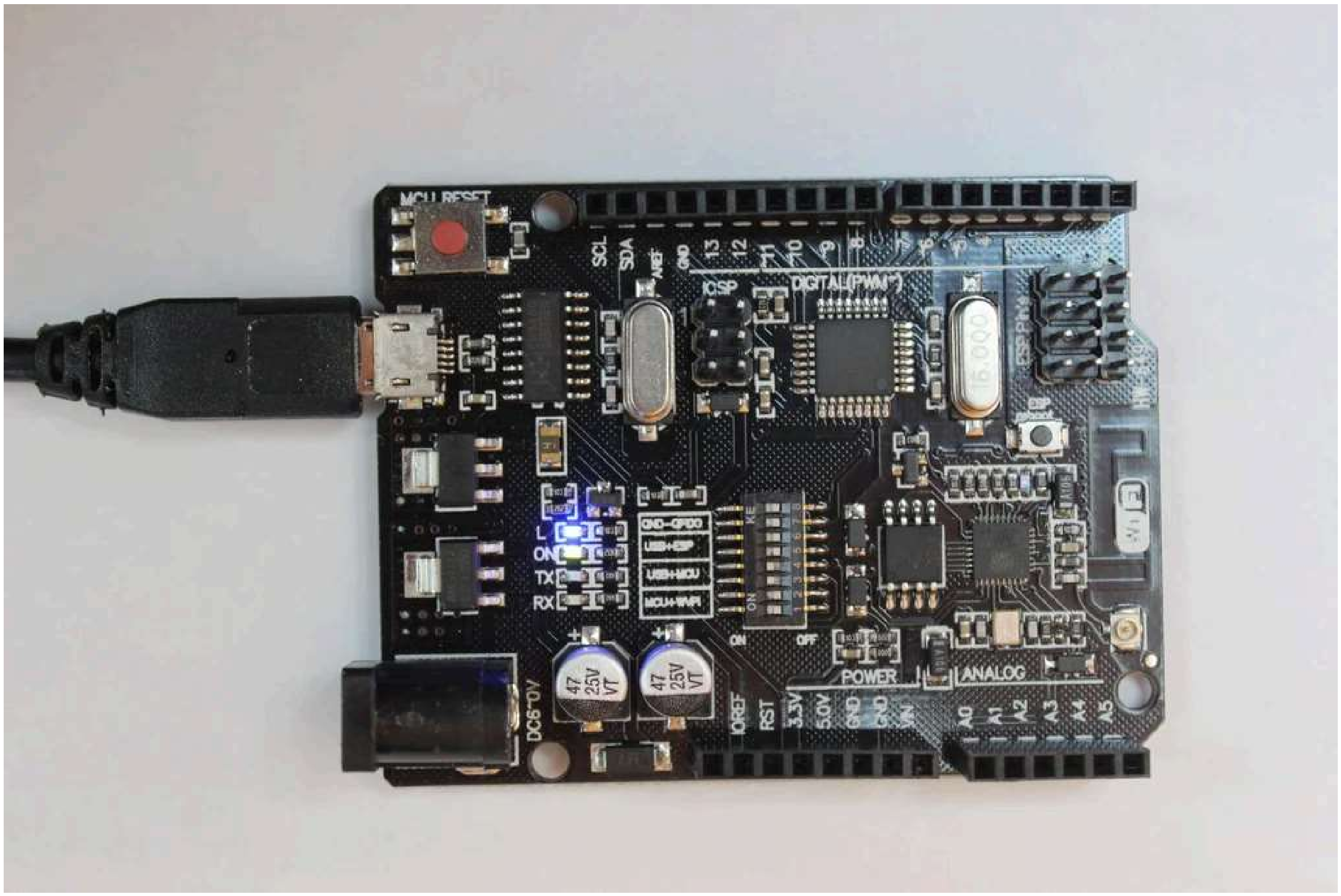
Mode 1 - USB to AT - SW3 and SW4 on. All other off

Mode 2 - USB to ESP Prog - SW5, SW6, SW7 on. All other off

Mode 3 - USB to ESP - SW5, SW6 on. All other off

Mode 4 - AT to ESP - SW1, SW2 on. All other off

Step 3: Programming the ATMega328

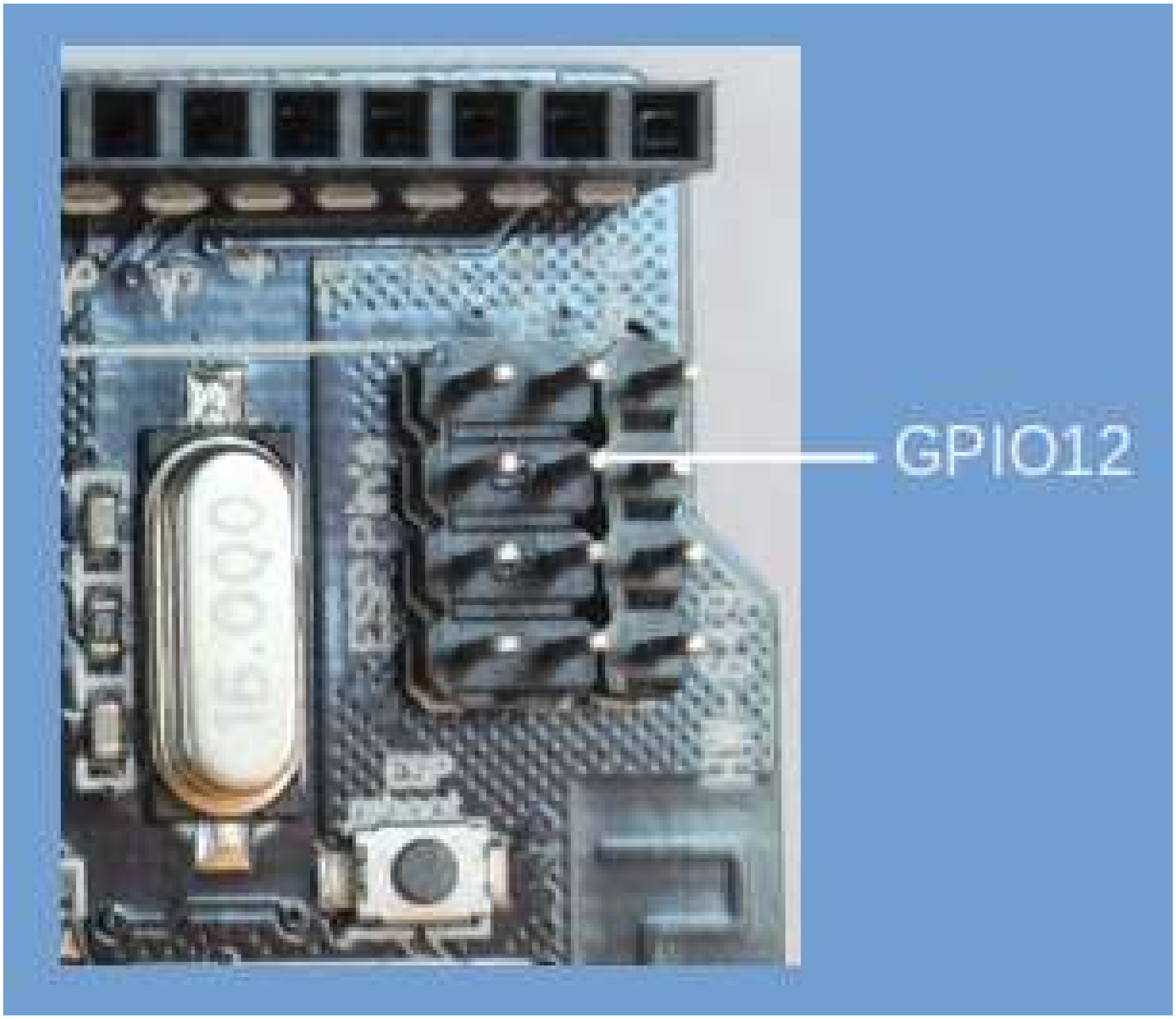


As a first demonstration, we can program the ATMega328 with the Blink Sketch. Steps as follows:

- Disconnect the board from power
- Using a paper clip or pencil point (or any other sharp end device), set the switches to Mode 1. Th is means that SW3 and SW4 are On, all other switches are off.
- Call up the Blink Sketch in the Arduino IDE from Examples → Basic
- Connect the Micro USB port to your computer with a USB to Micro USB cable
- Upload the Sketch
- The ATMega328 will automatically reset and the LED labeled “L” will start blinking.

Success, the ATMega328 is working and can be programmed. Simple.

Step 4: Programming the ESP8266



This part is a little more complicated.

Before programming the ESP8266, you have to ensure that the Arduino IDE you are using is configure d to do this type of programming. There are numerous articles on the Internet explaining how to do thi s. A search of “program ESP8266 from Arduino IDE” will provide many results.

Here are a few links:

https://www.instructables.com/Setting-Up-the-Arduino-IDE-to-Program-ESP8266/

https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/

The basic steps are as follows:

- In File → Preferences, add a url to the Additional Boards Manager: http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Under Tools → Boards → Boards Manager, Search for ESP8266 and install latest version of "ESP8266 by ESP8266 Community"
- Once this is done, you should have an option under Tools→ Board for a ESP8266 Boards. Choose Generic ESP8266 Module.

After all that you are ready to get a Blink program working on the board. There appears to be no built in LED for the ESP8266 on this board, so you have to connect an LED to demonstrate the blink program.

The ESP header on the board allows access to GPIO12 from the ESP8266, this will be used for connecting the LED. The pin functions are given on the rear of the board. Connect a 330 Ohm (or similar) and LED wired in series between GPIO12 and the GND pin on the Arduino header.

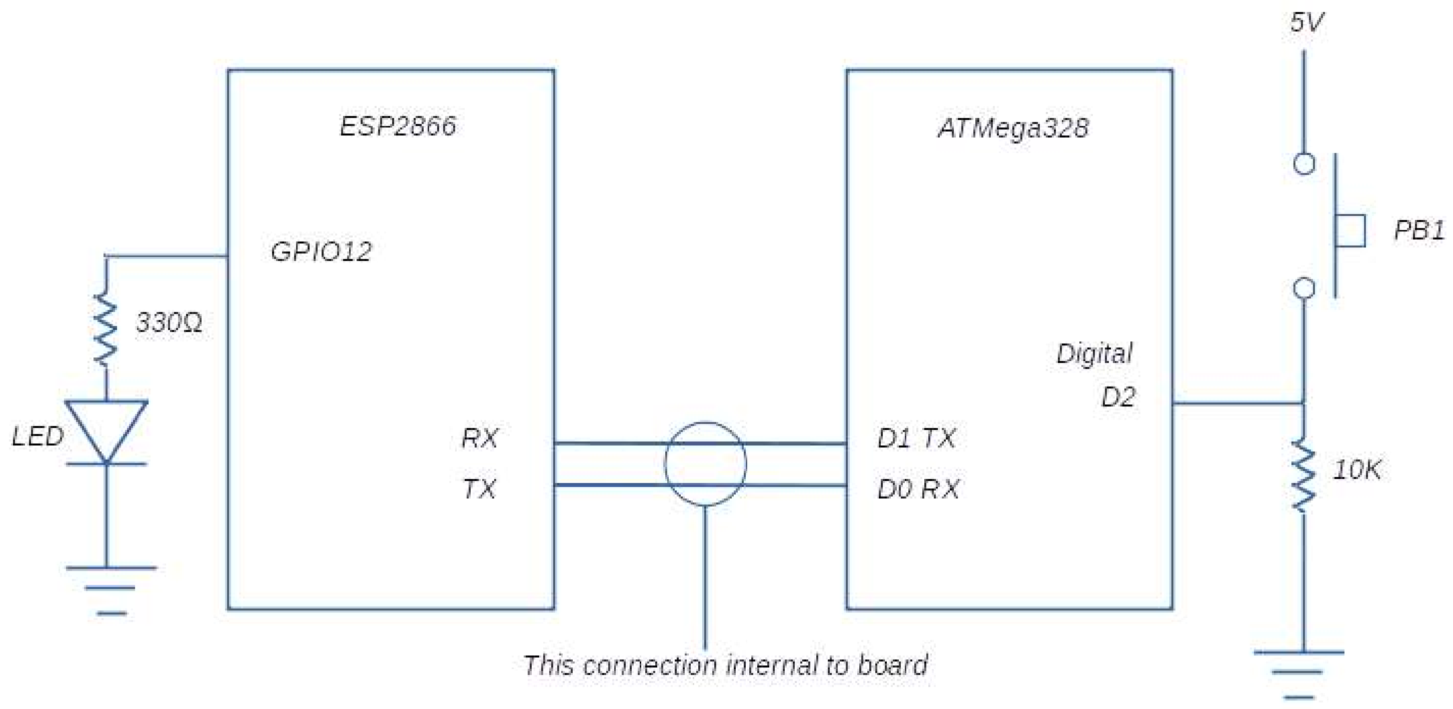
Use the following steps:

- Make sure the Arduino IDE is set for the Generic ESP8266 Module
- Remove power from the board. Set the switches to Mode 2. This means that SW5, SW6, SW7 are On. all other switches are Off.
- Connect the Micro USB port to your computer with a USB to Micro USB cable
- Upload the Sketch Blink ESP8266
- If successful, the Arduino IDE will produce the following message – "Hard resetting via the RTS pin"
- Remove power from the board and toggle SW7 to off. This takes the ESP8266 out of programming mode.

Restore power to the board and the LED connected to GPIO12 should start blinking. Success!

```
/*
  ESP8266 Blink
  Blink the LED on the ESP8266 module
  This example code is in the public domain

  The external LED on the ESP8266 module is connected to GPIO12

*/

void setup() {
  pinMode(12, OUTPUT);     // Initialize pin 12 which is GPIO12 as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(12, LOW);   // Turn the LED off
  // low is off on the ESP8266-12
  delay(1000);                      // Wait for a second
  digitalWrite(12, HIGH);           // Turn the LED on by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to distinguish between low and high)
}
```

Step 5: Communicating Between ATMega328 and ESP8266



The ATMega328 and ESP8266 can communicate via the serial bus include in both chips. This commu nication is bilateral; ATMega826 can send information to the ESP8266 and vice versa.

To demonstrate this, a simple circuit connected to the ATMega328 can activate an LED connected to t he ESP8266. Circuit diagram shown in this step. Once the circuit is completed, the programming steps as follows:

- Set the DIP switch to Mode 1 (SW3 and SW4 are On)with power removed. Restore power and u pload the Arduino Sketch “ArduinoESPCommunication
- Remove power and set the DIP switch to Mode 2 (SW5, SW6, SW7 are On)
- Change Arduino IDE to ESP8266 programming mode and upload the sketch “ESPRecieveCom munication”
- Remove power and set the dip switch to Mode 4 (SW1 and SW2 are On)
- Restore power
- Pressing the push button will cause the ATMega328 built in LED to flash and the LED connected to the ESP8266 will come on.
- Pressing the push button again will flash the built in LED and turn off the LED connected to the E SP8266

Communication between the ATMega328 and the ESP8266 uses the Serial.write(“change”) and Seria l.readString() commands. See the sketches for more details.

```
//ArduinoESPCommunication
const int buttonPin = 2;     // the number of the pushbutton pin
int buttonState = 0;
void setup() {

// Initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  Serial.begin(115200); //All ESP8266 serial communication at 115200 baud
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // Send a message to the ESP8266
    Serial.write("change");
    // Flash the built in LED to confirm that button push was registered
    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
    delay(500);
    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
    delay(250);

  }
  else {
    delay(25);
  }
}



//ESPRecieveCommunication
bool toggle;
String incomingString;

void setup() {
  pinMode(12, OUTPUT);     // Initialize pin 12 which is GPIO12 as an output
  toggle = false;

  Serial.begin(115200);   //All serial communication with ESP8266 at 115200 baud
}

void loop() {

  if (Serial.available() > 0){
    // read the incoming string from the Arduino
    incomingString = Serial.readString();

    // If the incoming string from the Arduino equals "change" then we toggle the LED
    if (incomingString.equals("change")){
      digitalWrite(12, toggle);  // Turn the LED off or on
      toggle = !toggle;
    }
  }
  delay(50); // Wait for half second
}
```

Step 6: Final Words

Not known why this specific development board is not more popular (based on the number of articles and documentation available on line). It has several advantages over trying to implement an Arduino w ith a separate ESP8266 board and provides dual processing power.

Good luck using this board!